

# CSS Cookbook

**Christopher Schmitt**

*Prólogo por Dan Cederholm*

Authorized Portuguese translation of the English edition of *CSS Cookbook*, 3E ISBN 978-0-596-15593-3 © 2009, O'Reilly Media Inc. This translation is published and sold by permission of O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

Tradução em português autorizada da edição em inglês da obra *CSS Cookbook*, 3E ISBN 978-0-596-15593-3 © 2009, O'Reilly Media Inc. Esta tradução é publicada e vendida com a permissão da O'Reilly Media, Inc., detentora de todos os direitos para publicação e venda desta obra.

© Novatec Editora Ltda. 2010.

Todos os direitos reservados e protegidos pela Lei 9610 de 19/02/1998.

É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, do autor e da Editora.

Editor: Rubens Prates  
Tradução: Rafael Zanolli  
Revisão gramatical: Márcio Friedl  
Editoração eletrônica: Camila Kuwabata

ISBN: 978-85-7522-230-0

Histórico de impressões:

Agosto/2010      Primeira edição

Novatec Editora Ltda.  
Rua Luís Antônio dos Santos 110  
02460-000 – São Paulo, SP – Brasil  
Tel.: +55 11 2959-6529  
Fax: +55 11 2950-8869  
E-mail: [novatec@novatec.com.br](mailto:novatec@novatec.com.br)  
Site: [www.novatec.com.br](http://www.novatec.com.br)  
Twitter: [twitter.com/novateceditora](https://twitter.com/novateceditora)  
Facebook: [facebook.com/novatec](https://facebook.com/novatec)  
LinkedIn: [linkedin.com/in/novatec](https://linkedin.com/in/novatec)

**Dados Internacionais de Catalogação na Publicação (CIP)**  
**(Câmara Brasileira do Livro, SP, Brasil)**

Schmitt, Christopher  
CSS cookbook / Christopher Schmitt ; [tradução  
Rafael Zanolli]. -- São Paulo : Novatec Editora ;  
Sebastopol, Calif. : O'Reilly, 2010.

Título original: CSS cookbook.  
ISBN 978-85-7522-230-0

1. CSS (Linguagem de programação) 2. XHTML  
(Linguagem de programação) 3. Web sites - Criação  
4. Web sites - Desenvolvimento 5. Web sites -  
Design I. Título.

10-08145

CDD-004.678

Índices para catálogo sistemático:

1. Web sites : Criação : Aplicação de CSS e  
(X)HTML : Ciência da computação 004.678

# Utilização básica do HTML

## 1.0 Introdução

Utilizar CSS com eficiência exige o uso efetivo da linguagem HTML. Para colocá-lo no caminho certo no trabalho com a linguagem HTML, este capítulo passará pelas noções básicas de sua utilização correta. Você explorará técnicas fundamentais e críticas para a criação de uma página HTML, para validação da linguagem de marcação certificando-se de que ela esteja livre de erros gerais e de digitação e para o uso de novas possibilidades, como a adição de áudio e vídeo que pode ser feita no HTML5.



Caso você já se considere um veterano nessa área, sinta-se livre para folhear o capítulo. Até mesmo uma revisão de seu conteúdo poderá ajudá-lo a criar alguns bons hábitos que facilitarão seu trabalho.

## Estrutura de documentos

Para criar o design de suas páginas, é preciso primeiro que haja conteúdo no documento web, normalmente um simples arquivo de texto. Esse conteúdo, dentro do arquivo, deve ter tags que respeitem o que é chamado de *HyperText Markup Language* (Linguagem de Marcação de Hipertexto), mais conhecida como HTML. A linguagem HTML oferece estrutura aos documentos, por meio do uso de *elementos*.

Quando você envolve esses elementos com tags em todo seu conteúdo, como `p` para parágrafos e `h2` para cabeçalhos, a página web começa a formar uma estrutura inerente de documento HTML.

O navegador, então, aplica sua própria *stylesheet* (folha de estilo) para formar o que se considera a exibição default da página web nessa estrutura de documento.

Essa aparência básica não vencerá prêmio de design algum, mas é um ponto de partida, permitindo que a *apresentação* ou design sejam associados de modo mais limpo por Cascading Style Sheets (CSS) e JavaScript, conferindo aparência e movimento à página web.

## Marcação semântica

Este capítulo é uma cartilha sobre como codificar HTML de modo semântico. A marcação semântica é a noção “radical” de que devemos empregar o elemento HTML adequado para seu conteúdo respectivo.

Por exemplo, para denotar um parágrafo, utilizamos a tag `p` no início e no fim do texto do parágrafo:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.</p>
```

## Evite a velha sopa de tags

A abordagem semântica ao HTML não é comum na web. Uma vez que vários elementos HTML têm aparências diferentes quando são exibidos por um navegador, web designers muitas vezes produzem misturas estranhas de elementos HTML, criando algo comumente conhecido como *sopa de tags*, preocupados apenas em atingir a aparência e a impressão desejadas.

Para assumir o controle desses fatores, os designers podem escolher adicionar tags HTML de apresentação a um conteúdo de outra forma semanticamente marcado, da seguinte maneira:

```
<p><font face="Georgia, Times, serif" size="+2">Lorem ipsum dolor</font>  
<font face="Arial, Helvetica, sans-serif" size="1">sit amet, consectetur adipiscing  
elit, sed diam <b>nonummy</b> nibh euismod tincidunt ut laoreet dolore magna  
aliquam erat volutpat.</font></p>
```

Todas essas tags adicionais estão presentes apenas para controlar a aparência e sensação de um *único* parágrafo.

Quando você utiliza uma codificação HTML tradicional, cada elemento da página de um site terá de ser codificado com elementos adicionais para criar as cores, as fontes, o alinhamento e o layout específicos que um designer desejar; tal processo é extremamente tedioso e tende a erros.

Imagine que você esteja projetando um website que contenha 20 páginas e deseje adicionar certos elementos de design, como cores, fontes, tamanho das letras e alinhamento ao site. Agora imagine ter de fazê-lo na manutenção de um website de mil páginas. Que tal um de 1 milhão de páginas?

Os documentos HTML de um site rapidamente ficam presos a códigos adicionais que tornam, tanto seu conteúdo quanto o código em si, praticamente ingovernáveis.

## HTML é estrutura de documentos

Em virtude disso, é tão importante acertar ao máximo a estrutura de seu documento quando se trata de HTML. Fazendo uso de uma codificação semântica e esbelta, desenvolvedores web economizam tempo de manutenção e ainda fornecem o framework sobre o qual as stylesheets poderão ser aplicadas.



Caso se considere entendido o bastante em HTML e HTML5, talvez você já saiba as informações contidas neste capítulo. Se esse for o caso, pule sua leitura.

### 1.1 Escolha do editor de texto

#### Problema

Você deseja escolher um editor de texto para marcação de conteúdo em HTML.

#### Solução

Inúmeras aplicações de software estão direcionadas à codificação em HTML. Algumas são gratuitas, enquanto outras exigem pagamento.

Alguns editores básicos de texto que já vêm pré-instalados com sistemas operacionais incluem:

- Bloco de Notas (Notepad, Windows).
- TextEdit (Mac OS).
- Gedit (Linux).

Veja alguns outros editores de texto gratuitos que ostentam mais funções:

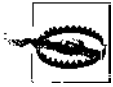
- Notepad++ (Windows; <http://notepad-plus.sourceforge.net/uk/site.htm>)
- TextWrangler (Mac OS; <http://www.barebones.com/products/TextWrangler/>)
- jEdit (Windows, Mac OS e Linux; <http://www.jedit.org/>)

Para ambientes de desenvolvimento integrados (Integrated Development Environments – IDE) mais profissionais e comerciais, tente um destes:

- Adobe Dreamweaver (Windows e Mac OS; <http://www.adobe.com/products/dreamweaver/>)
- Coda da Panic Software (Mac OS; <http://www.panic.com/coda/>)

## Discussão

Para edição de textos HTML, alguns aplicativos vêm junto de sistemas operacionais comuns, como o Mac OS X e o Windows. Eles são o TextEdit e o Bloco de Notas, respectivamente.



Não utilize processadores de texto para trabalhar com HTML. Ainda que esses programas sejam ideais para criação de documentos comuns que precisem ser impressos, eles adicionam formatações extrínsecas ao seu texto, as quais você não deseja, nem precisa.

Antes de utilizar o TextEdit, vá em **File** (Arquivo) → **Preferences** (Preferências) e selecione “Plain Text” (Texto Simples) ao lado da opção de formato. De outra forma, o editor de texto pode remover os elementos HTML.

Caso você esteja utilizando o Bloco de Notas, selecione **Format** (Formatar) → **WordWrap** (Quebra Automática de Linhas). Essa opção permite que linhas longas sejam quebradas dentro do aplicativo, tornando sua edição mais fácil.



Tanto para o TextEdit quanto para o Bloco de Notas, certifique-se de salvar o arquivo HTML com a extensão *.html*. Não coloque apenas uma extensão *.html* adicional ao fim do nome do arquivo. Por exemplo, *exemplo.txt.html* ou *exemplo.html.txt* levam apenas a desilusões.

Ainda que esses editores de código – que são gratuitos e já vêm instalados no sistema operacional – não ofereçam muitas opções, muitos web designers confiam neles para trabalhar com HTML.

### Mais robustos, porém ainda gratuitos

Outra opção para edição de textos que também é gratuita é o TextWrangler da Bare Bones Software. O TextWrangler não é tão completo quanto o principal produto da empresa, o BBEdit, mas pode da mesma forma satisfazer suas necessidades. Tanto o TextWrangler quanto o BBEdit são aplicativos apenas para Mac.

Para o Windows, há opções como o Notepad++ e o TextPad (<http://www.textpad.com/>).

Caso você utilize Unix, há os editores vi/vim e editores Emacs. Outro editor de texto potencial é o jEdit, que também está disponível para Mac e Windows.

### Soluções IDE

Produtos mais robustos muitas vezes custam mais, entretanto oferecem uma solução completa para a manipulação de quase todos os aspectos da criação de websites. Produtos populares desse tipo incluem o Adobe Dreamweaver e o Coda da Panic Software.

## Veja também

<http://www.notepad.org/logo.htm>, para conseguir um banner gráfico "Made with Notepad" (Feito com o Bloco de Notas) que você poderá colocar em sua página web.

## 1.2 Codificação de uma página HTML básica

### Problema

Você deseja criar sua primeira página HTML.

### Solução

Comece com o conteúdo básico, da seguinte forma:

```
My Basic Web Page
Epsum factorial non deposit quid pro quo hic escorol.
```

A seguir, adicione um elemento `html` em torno de todo o documento:

```
<html>
My Basic Web Page
Epsum factorial non deposit quid pro quo hic escorol.
</html>
```

Então, coloque os elementos `head` e `body` no documento, dessa forma:

```
<html>
  <head>
  </head>
  <body>
My Basic Web Page
Epsum factorial non deposit quid pro quo hic escorol.
  </body>
</html>
```

Insira um elemento `title` ao elemento `head`:

```
<html>
  <head>
    <title>CSS Cookbook</title>
  </head>
  <body>
My Basic Web Page
Sed quis custodiet ipsos custodes?
  </body>
</html>
```

Os elementos de cabeçalho (`h1`) e parágrafo (`p`) ficam no elemento `body`, e a página deverá ser exibida da mesma forma que a mostrada na figura 1.1.

```
<html>
  <head>
    <title>CSS Cookbook</title>
  </head>
  <body>
    <h1>My Basic Web Page</h1>
    <p>Sed quis custodiet ipsos custodes?</p>
  </body>
</html>
```

## Discussão

Toda página web precisa ter um elemento HTML que envolva todo o documento. Em cada elemento HTML, há dois elementos necessários: `head` e `body`.

O elemento `head` contém as informações sobre o documento, muitas vezes chamadas de metainformações. Ele deve conter o elemento `title` dentro de sua seção. Esse texto normalmente ocupa a porção de topo da janela do navegador e é utilizado na criação de bookmarks. É importante que se seja conciso e que se evitem longas descrições ao utilizar-se a tag `title`.

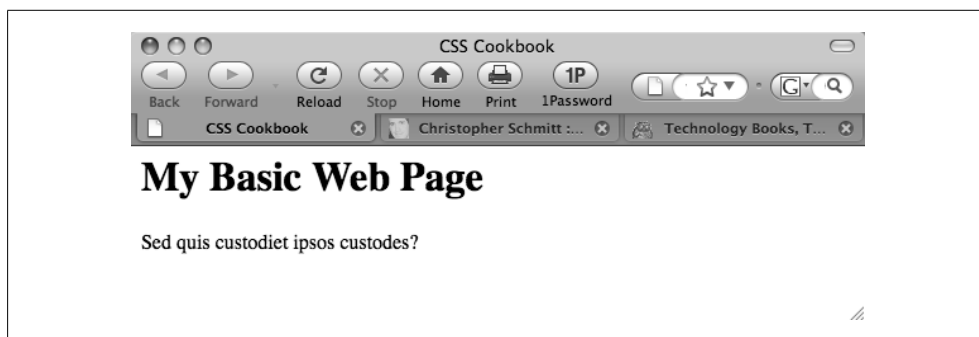


Figura 1.1 – Exibição padrão de uma página HTML básica.

Caso o elemento `title` não contenha texto, os navegadores utilizarão o nome do arquivo ou as primeiras palavras do documento em seu lugar.



Apenas texto é permitido no elemento `title`. Outros elementos HTML não são permitidos.

O conteúdo de um documento web é colocado no elemento `body`. Caso tenha de editar ou revisar uma página web, na maior parte do tempo você trabalhará dentro desse elemento.

Nesse exemplo, o título foi definido com um elemento `h1` ao lado do elemento padrão `p` para parágrafos.

## Veja também

A receita 1.1, para escolha de um editor de texto.

## 1.3 Compreensão de DOCTYPEs e efeitos no layout do navegador

### Problema

Você deseja tornar sua página válida e compatível com os padrões web.

### Solução

O HTML 4.01 oferece três tipos de documentos: *Strict* (Estrito), *Transitional* (Transicional) e *Frameset*.

Tanto o HTML5 quanto o XHTML 1.1 oferecem um único tipo de documento, no entanto o XHTML 1.0, da mesma forma que o HTML 4.01, oferece três.

Apenas *uma* definição de tipo de documento (document type definition ou DTD) aparece no documento HTML. Utilize qualquer um dos seguintes DOCTYPEs que melhor se encaixe com as necessidades de seu projeto.

HTML 4.01 Strict DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

HTML 4.01 Transitional DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/1999/REC-html401-19991224/loose.dtd">
```

HTML 4.01 Frameset DTD:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
" http://www.w3.org/TR/1999/REC-html401-19991224/frameset.dtd">
```

HTML5 DTD:

```
<!DOCTYPE html>
```

XHTML 1.0 Strict DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

XHTML 1.0 Transitional DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

XHTML 1.0 Frameset DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

XHTML 1.1 DTD:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

Veja uma página básica com o DTD do HTML5 e os elementos necessários `head`, `body` e `HTML`:

```
<!DOCTYPE html>
<html>
  <head>
    <title>CSS Cookbook</title>
  </head>
  <body>
    <h1>My Basic Web Page</h1>
    <p>Epsum factorial non deposit quid pro quo hic escorol.</p>
  </body>
</html>
```

## Discussão

Um `DOCTYPE`, abreviação de *document type definition* (definição de tipo de documento), define os blocos construtores em HTML ou XHTML de um documento e informa aos navegadores e validadores qual versão de HTML ou XHTML seu documento utiliza.

A declaração `DOCTYPE` deve aparecer no início do documento de cada página, antes do elemento `html`, para garantir que sua marcação e CSS sejam compatíveis com os padrões web e que os navegadores lidem com a página de acordo com os DTDs apropriados.

## Modo quirks

O XHTML exige um `DOCTYPE` válido no topo do documento; de outra forma, as páginas não serão validadas, e os navegadores cairão no que é conhecido como *modo quirks*.

O modo quirks ocorre quando um navegador trata uma página web como uma página com “bugs”. Como resultado, tais páginas são tratadas como se tivessem sido escritas com marcação inválida e serão renderizadas incorretamente nos navegadores modernos mesmo que o XHTML e as CSS tenham sido codificados perfeitamente.

Uma página web que não tem um `DOCTYPE`, com um `DOCTYPE` antigo, ou com um `DOCTYPE` que contém erros de digitação, pode disparar o modo quirks. Assim, ao codificar suas páginas, certifique-se de que o `DOCTYPE` tanto foi adicionado quanto foi digitado corretamente, a fim de garantir que os navegadores não renderizem suas páginas no modo quirks.



Caso uma página web tenha um DOCTYPE HTML5, os navegadores modernos iniciarão o modo Standards, mesmo que a marcação em si não tenha sido codificada com elementos HTML5. As versões 6 e 7 do Internet Explorer ignoram atributos HTML5.

As figuras 1.2 e 1.3 mostram como uma tabela contida em um elemento `div` com uma largura de 100% entra no modo quirks no Internet Explorer 6 e qual seria a aparência da página no modo standards.

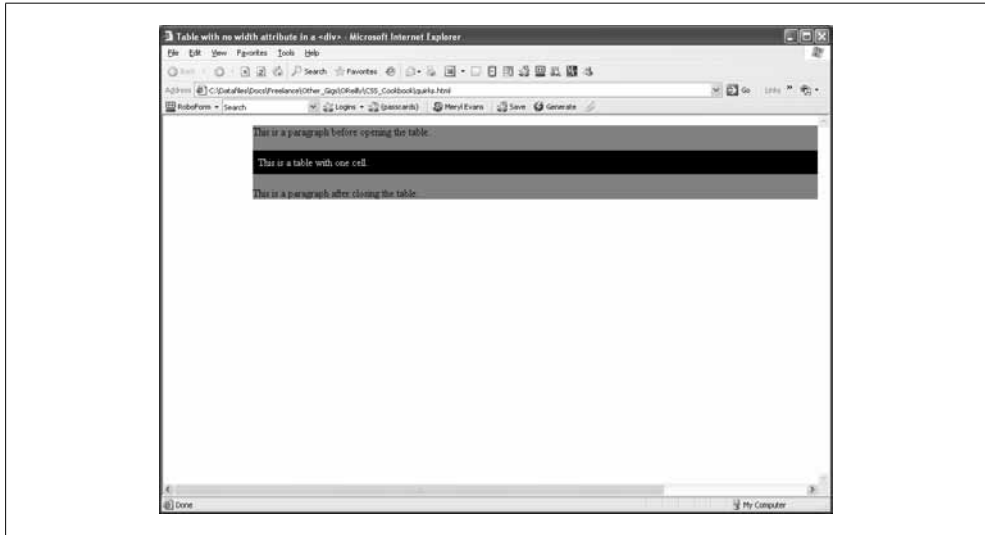


Figura 1.2 – Largura de tabela no Internet Explorer 6 no modo quirks com nenhum DOCTYPE incluso.

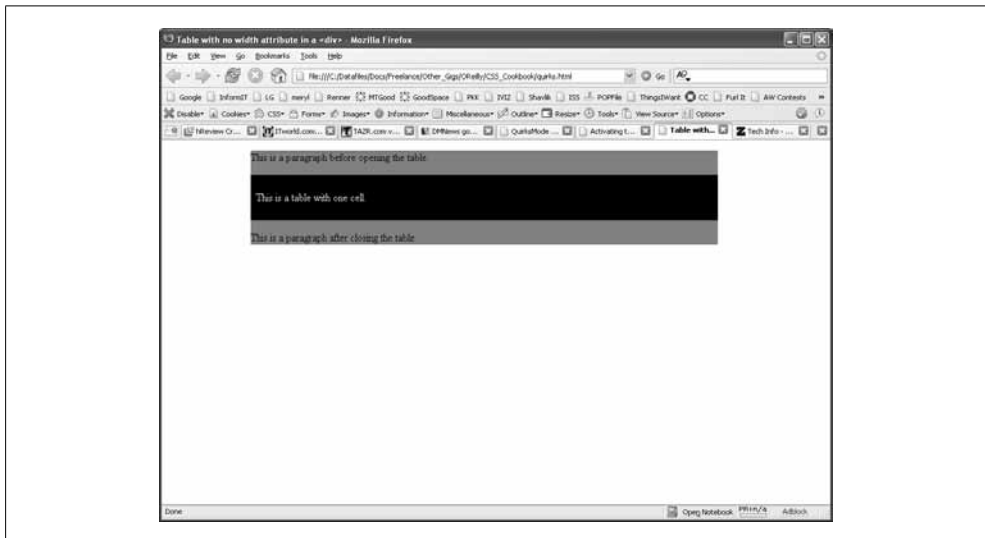


Figura 1.3 – Largura de tabela no Firefox 1.5 no modo standards com DOCTYPE HTML 4.01 Strict.

## Por que não utilizar o DOCTYPE mais recente?

Utilizar DOCTYPEs mais novos como o HTML5 é uma opção, porém não é a única. Diferentemente de lançamentos de aplicativos, os novos DOCTYPEs não tornam os mais antigos irrelevantes.

Por exemplo, seria muito difícil instalar e (mais difícil ainda) rodar o Photoshop 4 nos computadores de hoje. Todavia, você ainda pode utilizar a sintaxe e os DOCTYPEs HTML4 sem medo de que os navegadores não renderizem seu conteúdo.

## A menor página web de todos os tempos

A solução oferece um exemplo de uma página HTML5 relativamente pequena. Entretanto, um exemplo ainda menor e válido pode ser utilizado:

```
<!DOCTYPE html>
<title>Small HTML5</title>
<p>Hello world</p>
```

Esses três elementos HTML podem ser validados para HTML5 conferindo a página em <http://validator.w3.org/check?uri=http%3A%2F%2Fjsbin.com%2Fowata&ss=1>.

## Veja também

A especificação HTML5 para DTD, em <http://dev.w3.org/html5/spec/Overview.html#the-doctype>; a especificação HTML 4.01 para DTD, em <http://www.w3.org/TR/html401/intro/sgmltut.html#h-3.3>; os validadores do W3C, em <http://www.w3.org/QA/Tools/#validators>; um artigo sobre DOCTYPEs no A List Apart em <http://www.alistapart.com/articles/doctype/>; um artigo do QuirksMode em <http://www.quirksmode.org/index.html?css/quirksmode.html>; a informação da Mozilla sobre o modo quirks, explicando as diferenças entre os modos de renderização e como eles lidam com o modo quirks, em [https://developer.mozilla.org/en/Mozilla's\\_Quirks\\_Mode](https://developer.mozilla.org/en/Mozilla's_Quirks_Mode); e a página DOCTYPE do Opera, em <http://www.opera.com/docs/specs/doctype/>

## 1.4 Marcação de cabeçalhos

### Problema

Você deseja diferenciar a importância dos títulos dentro de um mesmo documento.

### Solução

Utilize um dos seis títulos disponíveis, h1 a h6, como mostrado na figura 1.4:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
```

```
<head>
  <title>CSS Cookbook</title>
</head>
<body>

  <h1>My Basic Web Page</h1>
  <p>Epsum factorial non deposit quid pro quo hic escorol.</p>

  <h2>Secondary Heading</h2>
  <p>Feles mala! cur cista non uteris? stramentum novum in ea posui.</p>

  <h3>Tertiary Heading</h3>
  <p>Por scientie, musica, sport etc., li tot Europa usa li sam vocabularium.</p>

  <h4>Quaternary Heading</h4>
  <p>Lex clavatoris designati rescindenda est.</p>

  <h5>Quinary Heading</h5>
  <p>Ire fortiter quo nemo ante iit.</p>

  <h6>Senary Heading</h6>
  <p>Interdum feror cupidine partium magnarum europe vincendarum.</p>

</body>
</html>
```

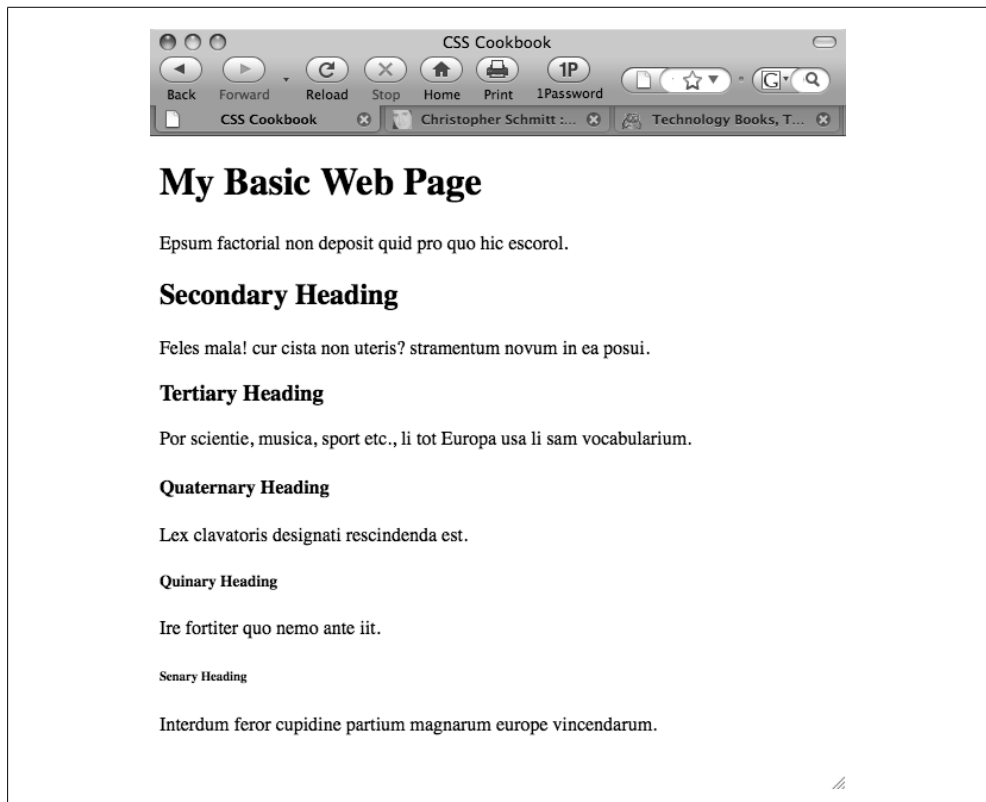


Figura 14 – Renderização padrão de seis níveis de títulos.

## Discussão

Você pode escolher entre os seis diferentes níveis quando realizar a marcação de títulos de um documento.

Ao marcar o conteúdo, certifique-se da utilização dos títulos na ordem correta. Por exemplo, caso você utilize o elemento `h2`, o título a seguir dele deve ser encapsulado pelo elemento `h3` (não `h4` ou `h5`). O cabeçalho da página não deve ser envolto pelo elemento `h2` (utilize o elemento `h1`). Em resumo, não pule suas tags de cabeçalho!

Não é importante que se utilizem *todos* os cabeçalhos na criação de um documento. No entanto, não exagere na utilização do elemento `h1`, uma vez que isso poderá diminuir sua classificação nos mecanismos de busca. Utilize o elemento `h1` uma vez para o título único de sua postagem ou página; em seguida, utilize o `h2` e `h3` para outras partes do documento.



Caso você tenha de utilizar os elementos `h4`, `h5` e `h6` em seu documento, quebre o conteúdo em páginas separadas ou investigue sua estrutura. Um documento que requer seis níveis diferentes de títulos pode estar tão carregado de conteúdo que não conseguirá reter a atenção média de uma pessoa.

Assim, não há motivos para se preocupar com a aparência dos títulos, pois com o poder das CSS, seu design (bem como todo o resto da página) pode ser modificado.



O uso correto de cabeçalhos em um documento beneficia as pessoas que estejam utilizando leitores de tela. Para uma demonstração, veja o vídeo em [http://www.youtube.com/watch?v=AmUPhEVWu\\_E](http://www.youtube.com/watch?v=AmUPhEVWu_E).

## Veja também

O capítulo 3 para a modificação de cabeçalhos e outros tratamentos de tipo comuns.

## 1.5 Realização de citações adequadas

### Problema

Você deseja fornecer citações com HTML, conforme a figura 1.5.

### Solução

Utilize o elemento `blockquote` ao citar uma grande quantidade de texto:

```
<blockquote cite="http://www.example.com/">  
  <p>Si fallatis officium, quaestor infitias eat se quicquam scire de factis vestris.</p>  
</blockquote>
```

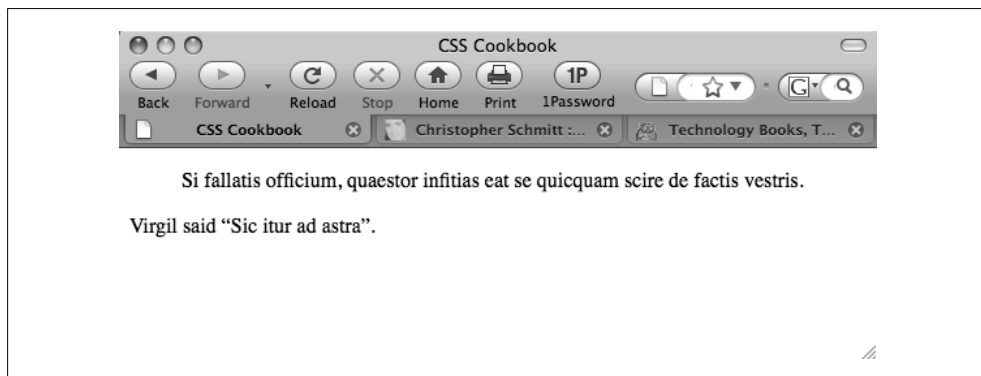


Figura 1.5 – Renderização padrão de citações.

Para citar frases, utilize o elemento `q`:

```
<p>Virgil said <q>Sic itur ad astra</q>.</p>
```

## Discussão

O elemento `blockquote` é um elemento *nível de bloco*. Isso significa que o texto marcado com um elemento `blockquote` separa-se do resto do texto forçando uma quebra de linha acima e outra abaixo de si.

O elemento `q` é um elemento *inline* (em linha), o qual não força uma quebra de linha. Elementos inline são úteis para citação de pequenas porções de texto dentro de um elemento parágrafo.



O elemento `q` é tipicamente renderizado com pontos de interrogação ao redor do texto que ele envolve. Todavia, essas interrogações não aparecem no Internet Explorer para Windows.

O atributo `cite` é opcional para os elementos `blockquote` e `q`. O valor de um atributo `cite` é a URL da qual a citação se origina.

## Veja também

O capítulo 3 para tratamentos de tipo comuns.

## 1.6 Adição de uma imagem

### Problema

Você deseja adicionar uma imagem a uma página web, conforme a figura 1.6.



Figura 1.6 – Imagem inserida em um documento.

## Solução

Utilize o elemento `img` para especificar a localização do arquivo de imagem:

```

```

Adicione um atributo `alt` dentro do elemento `img` para fornecer um texto alternativo caso as imagens estejam desativadas, ou as pessoas estejam navegando com uma tecnologia assistiva, como um leitor de telas:

```

```

## Discussão

O elemento `img` não se dirige a conteúdo dentro do documento web em si. Ele meramente define a localização de seu posicionamento dentro do documento e especifica sua localização relativa ao documento HTML.

## Dicas adicionais

Mesmo que uma imagem valha por mil palavras, o valor do atributo `alt` deve consistir de uma descrição relativamente breve.

Como mostrado na figura 1.6, alguns navegadores exibirão ao lado do cursor a chamada *tool tip*, mostrando o texto contido no atributo `title` de uma imagem:

```

```

## Formatos de arquivos

Formatos comuns de imagens suportados pelos navegadores incluem GIF e JPEG. Ambos formatos têm seus próprios prós e contras no que se refere a quais tipos de imagens são ideais para cada tipo.

De acordo com o *esquema de compressão*, o método pelo qual o tamanho de imagem é reduzido, GIFs são melhores para áreas de cores planas e com menor gradiente, enquanto JPEGs são bons para fotos e alterações sutis de cores.

Todos os navegadores aceitam o formato PNG; porém a transparência alpha é apenas aceita no Internet Explorer 8. Essa transparência permite opacidade, ou níveis de transparência, dentro de uma imagem, diferente do formato GIF, que pode determinar a transparência apenas de uma única cor. Caso uma versão mais antiga do IE renderize uma imagem PNG com transparência alpha, as porções transparentes normalmente se transformarão em blocos sólidos de cor branca.

## Diferenciação entre letras maiúsculas e minúsculas

Ao especificar um arquivo de imagem dentro do HTML, certifique-se de que o nome do arquivo não contém espaços e de que os caracteres maiúsculos e minúsculos correspondem. Ainda que o sistema operacional de seu computador possa não se incomodar com essa diferença, é provável que o servidor web hospedando seus arquivos tenha problemas, o que poderá impedir a exibição de suas imagens no navegador.

## Veja também

O capítulo 4 para o projeto de páginas web com imagens.

## 1.7 Adição de áudio com HTML5

### Problema

Você deseja adicionar áudio a uma página web com HTML5.

## Solução

Utilize o elemento `audio` para especificar um arquivo de áudio, conforme mostrado na figura 1.7:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>CSS Cookbook</title>
  </head>
  <body>
    <h1>Audio Example</h1>
    <audio src="html5audio.ogg" autoplay controls>
      <a href="html5test.ogg">Download audio</a>
    </audio>
  </body>
</html>
```

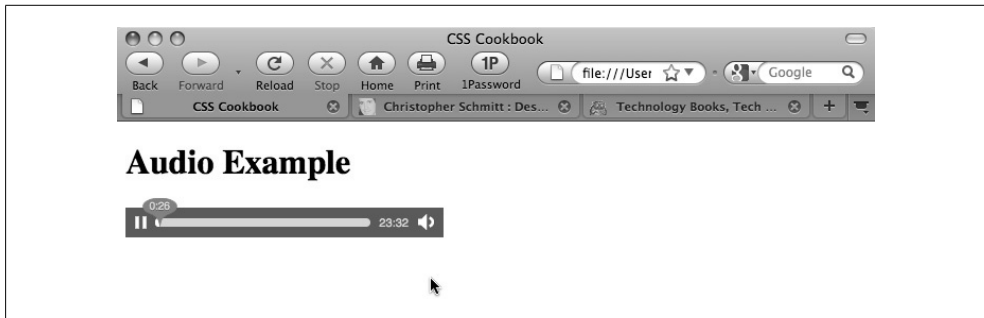


Figura 1.7 – Áudio adicionado a uma página web.

## Discussão

O elemento `audio` conta cinco atributos associados a ele: `src`, `autobuffer`, `autoplay`, `loop` e `controls`. Caso você não apresente o atributo `controls`, o reprodutor de áudio desaparece.

## Compatibilidade de áudio

No momento da publicação deste livro, não há um reprodutor de áudio que funcione em todos os navegadores que aceitam o elemento `audio` do HTML5, como mostrado pela tabela 1.1.

Tabela 1.1 – Suporte aos formatos de arquivo de áudio no HTML5

	Firefox 3.5	Safari 4	Chrome 3 beta	Opera 10
Ogg Vorbis	Sim		Sim	
MP3		Não	Não	
WAV	Sim	Sim		Sim

Para criar uma solução para muitos navegadores, utilize o elemento `audio` em conjunto com o elemento `source` que cita ambos os formatos OGG e MP3. Introduza em seguida o código `embed` e `object` do Flash Player:

```
<audio controls autobuffer>
  <source src="html5audio.ogg" />
  <source src="html5audio.mp3" />
  <!-- inclua aqui o código EMBED e OBJECT do Adobe Flash Player -->
</audio>
```



Se inserir o áudio, não é recomendada a configuração do arquivo em `autoplay`, uma vez que isso interfere na experiência dos usuários que estejam utilizando leitores de tela.

## Veja também

Receita 1.8 para adição de vídeo a páginas web.

## 1.8 Embutir vídeo com HTML5

### Problema

Você deseja adicionar vídeo ao HTML5.

### Solução

Utilize o elemento `video` do HTML5, como mostrado na figura 1.8:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>CSS Cookbook</title>
  </head>
  <body>
    <h1>Video Example</h1>
    <video src="html5video.ogg" width="320" height="240"
      controls poster="html5video.jpg">
      <a href="html5video.ogg">Download movie</a>
    </video>
  </body>
</html>
```

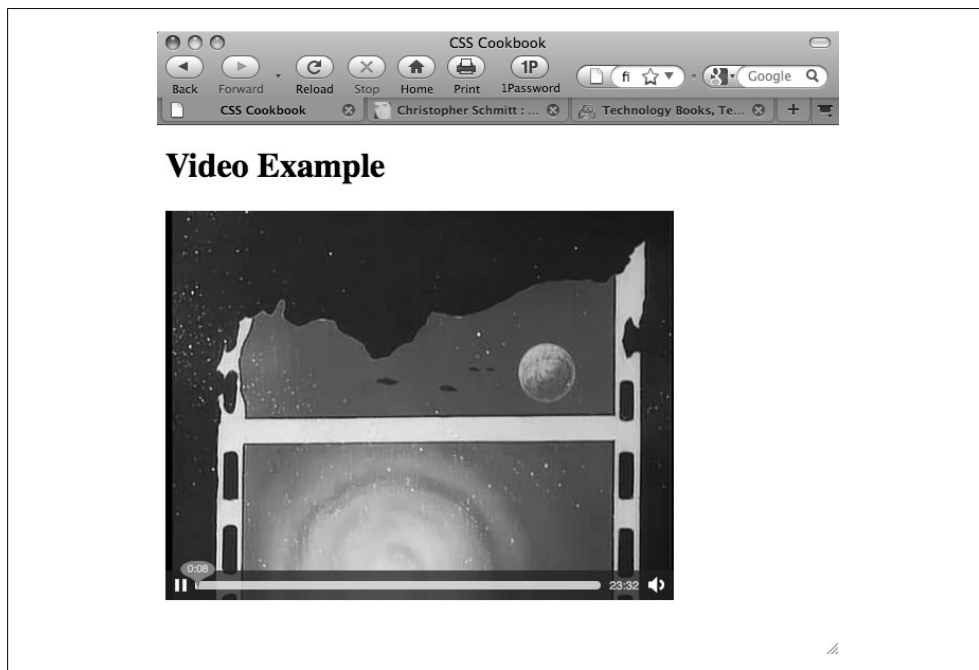


Figura 1.8 – Vídeo adicionado a uma página web.

## Discussão

Não é preciso especificar largura e altura do elemento `video`. Caso você não o defina com seus atributos respectivos, o filme será exibido com os valores padrão do arquivo de vídeo.

Um arquivo de vídeo pode conter seu próprio *poster*, uma imagem estática que representa o vídeo como um todo, semelhante a uma imagem em miniatura. Entretanto, você pode fazer o override desse poster utilizando o atributo `poster`. A imagem poster pode ser de qualquer tipo de arquivo suportado pelo navegador (p. ex., GIF, JPEG ou PNG).



Ainda que o atributo `controls` seja opcional, pensando na usabilidade, recomendo sua utilização de modo que não ofenda aos visitantes de seu site.

Você pode colocar texto alternativo entre as tags `video`, incluindo um link de download do arquivo para navegadores que não reconheçam o elemento `video`. Esse método confere a seus visitantes uma forma de visualizar o conteúdo com soluções de terceiros diferentes dos navegadores.

À época da publicação deste livro, o Safari 3.1 e posteriores, o Firefox 3.5 e posteriores, o Opera 10 beta e o Chrome 3 beta, todos oferecem suporte ao elemento `video`.

## Veja também

<http://www.videolan.org/> para informações sobre as ferramentas de exportação contidas no aplicativo VLC, que podem ser utilizadas na conversão de arquivos comuns de vídeo ao formato OGG (aceito pelo Firefox e Opera).

## 1.9 Uso efetivo do strong e do em

### Problema

Você deseja enfatizar certas palavras ou frases de um parágrafo, conforme mostrado na figura 1.9.

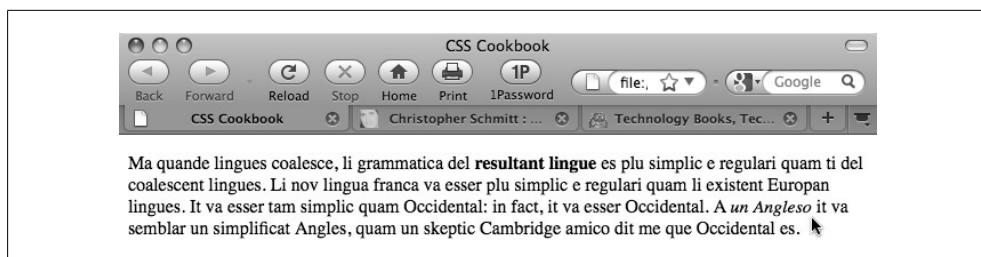


Figura 1.9 – Renderização padrão do texto destacado.

### Solução

Utilize os elementos `strong` e `em` para denotar ênfase dentro de um documento:

```
<p>Ma quande lingues coalesce, li grammatica del <strong>resultant lingue</strong>
es plu simplic e regulari quam ti del coalescent lingues. Li nov lingua franca
va esser plu simplic e regulari quam li existent European lingues. It va esser
tam simplic quam Occidental: in fact, it va esser Occidental. A <em>un
Angleso</em> it va semblar un simplifcat Angles, quam un skeptic Cambridge amico
dit me que Occidental es.</p>
```

### Discussão

A renderização padrão do elemento `strong` é colocar o texto em negrito, enquanto o elemento `em` define o texto em itálico.

Você utilizaria o `em` para chamar a atenção ou contrastar uma ou mais palavras do resto da frase. Por exemplo:

- *Darth Vader* pode ser traduzido de forma livre como *Pai das Trevas* em Holandês.
- Existem, não 57, mas 50 estados nos Estados Unidos da América.
- Caso você se junte a ele, ele *completará* seu treinamento.

O `strong` é um elemento alternativo ao `em` para chamar atenção a palavras ou frases.

Ainda que a utilização do `em` e do `strong` ajude a quebrar a monotonia de um texto, certifique-se de utilizá-las apenas de forma esporádica e consistente para que você não corra o risco de abusar de sua importância.

## Veja também

O capítulo 3 para tratamentos de tipo comuns.

## 1.10 Criação de listas

### Problema

Você deseja criar uma lista de itens dentro de uma página web, conforme a figura 1.10.

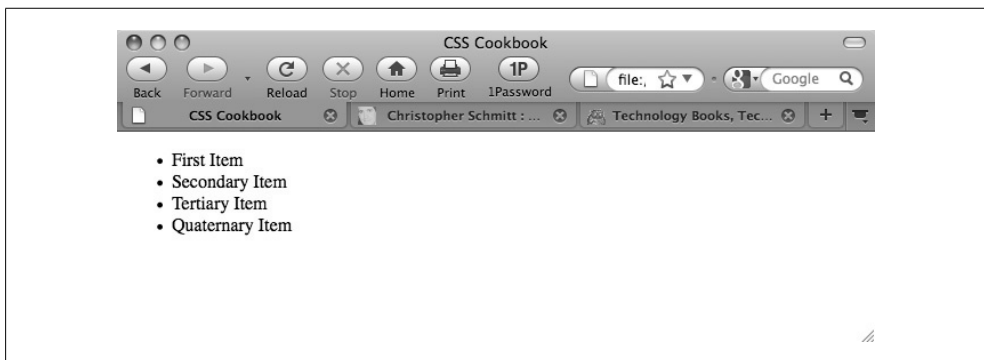


Figura 1.10 – Renderização padrão de uma lista não ordenada.

### Solução

Utilize o elemento `ul` para envolver uma lista de frases:

```
<ul>
  First Item
  Secondary Item
  Tertiary Item
  Quaternary Item
</ul>
```

Em seguida, utilize o elemento `li` para envolver cada item dentro da lista:

```
<ul>
  <li>First Item</li>
  <li>Secondary Item</li>
  <li>Tertiary Item</li>
  <li>Quaternary Item</li>
</ul>
```

## Discussão

Há três tipos de listas em HTML: listas *não ordenadas*, *ordenadas* e *de definição*.

A marcação de listas não ordenadas e ordenadas é bem simples e direta. Utilize dois elementos, `ul` e `li`, para marcar uma série de itens em uma lista não ordenada, a qual tipicamente resulta na colocação de um círculo do lado esquerdo de cada item da lista.



Uma lista não ordenada é geralmente utilizada na criação da base de um menu de navegação.

Listas ordenadas, que utilizam o elemento `ol` em lugar do elemento `ul`, têm uma ordem numérica sequencial adicionada ao começo de cada item.

Conforme mostrado na figura 1.11, listas de definição, que são utilizadas para definir termos, funcionam de modo um pouco diferente das listas ordenadas e não ordenadas. Cada item é dividido em duas partes: o termo (`dt`) e a definição (`dd`).

```
<dl>
  <dt>First Term</dt>
  <dd>Seriatim</dd>
  <dt>Secondary Term</dt>
  <dd>Sequentia</dd>
  <dt>Tertiary Term</dt>
  <dd>Sequens mirabitur aetas</dd>
</dl>
```

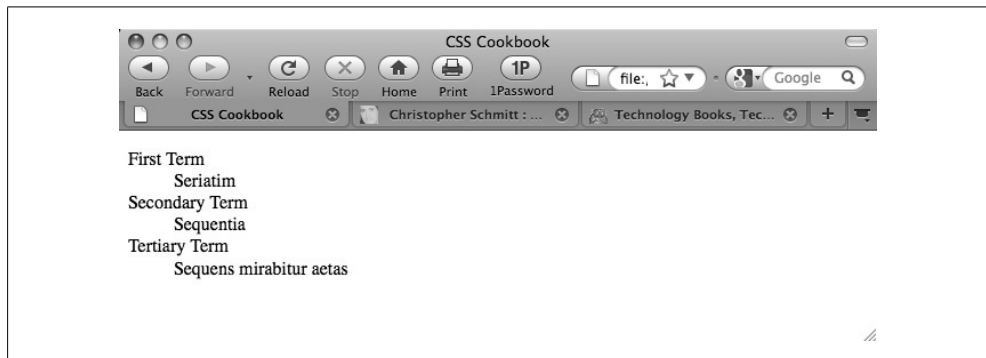


Figura 1.11 – Renderização padrão de uma lista de definição.

## Veja também

O capítulo 6 sobre listas e o capítulo 7, sobre links e navegação.

## 1.11 Criação de link para uma página web

### Problema

Você deseja oferecer um link a outra página web.

### Solução

Utilizando o link âncora:

```
<p>This book's <a href="http://www.csscookbook.com/">Web site</a> contains  
links to download more materials.</p>
```

para oferecer links a outra página do mesmo website, forneça o link de seu arquivo:

```
<p>Check out the <a href="about.html">About page</a> for more information.</p>
```

### Discussão

Junto ao elemento `img` (Receita 1.6) alguns navegadores exibem uma tool tip caso um atributo título e seu valor estejam presentes no link âncora, como mostra a figura 1.12:

```
<p>This book's <a href="http://www.csscookbook.com/" title="Link to the book site">  
Web site</a> contains links to download more material.</p>
```

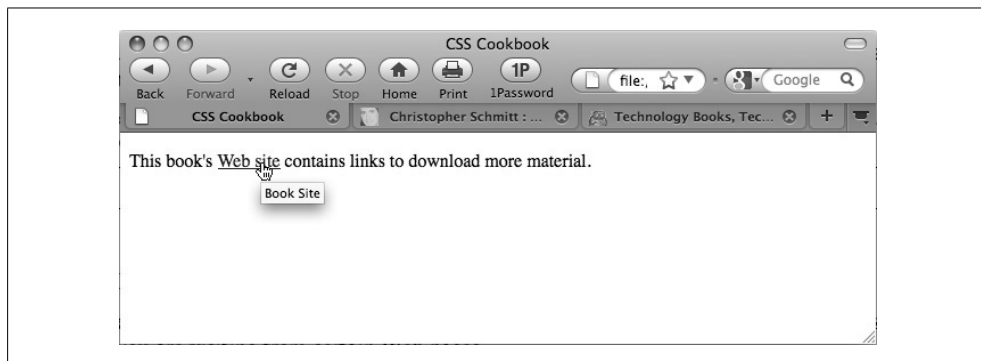


Figura 1.12 – Tool tip exibida sobre um link.

### Como oferecer um link para outra página dentro do mesmo site

Para criar links dentro do mesmo site, devem-se utilizar *links relativos* em lugar de links âncora. Links relativos são endereços que são válidos apenas se você estiver visitando a partir de determinadas páginas web.

Por exemplo, suponha que você tenha um website com quatro páginas dentro da mesma *pasta raiz*, o diretório que contém os arquivos do website, conforme mostrado na figura 1.13:

- *httpdocs/*
  - *index.html*
  - *aboutus.html*
  - *contactus.html*
  - *services.html*

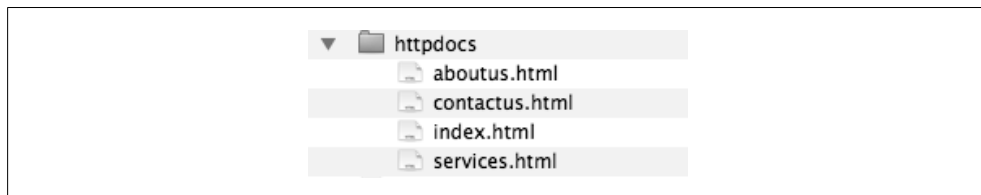


Figura 1.13 – Exemplo de estrutura do diretório.

Incluir em um link tudo que é necessário para apontar um navegador a uma localização significa criar um *link absoluto*, que deverá ter essa aparência:

```
<a href="http://www.csscookbook.com/services.html">Services Page</a>
```

Caso você deseje criar um link da página de índice para outra página dentro do mesmo website, é preferível que você utilize um *link relativo*, visto que é um pouco mais esbelto do que um link absoluto e, como nesse exemplo, pode citar apenas o nome do arquivo em si, dentro do atributo href:

```
<a href="services.html">Services Page</a>
```

Links relativos não contêm o protocolo *http://* completo nem o nome de domínio.

Quando acessa um link relativo, um navegador utiliza o nome de domínio da página que está atualmente exibindo na criação do link para onde deve se dirigir.

### Subindo pastas

Da mesma forma que seu computador pessoal pode conter inúmeras pastas com muitos arquivos de projetos, os websites também possuem uma série de pastas e arquivos. Para oferecer links de um documento a outro dentro de um mesmo website, utilize links relativos.

Por exemplo, digamos que você possui uma página principal de especificações dentro de uma pasta *specs*, a qual por sua vez está incluída em uma pasta *widget*. A organização dos arquivos no servidor pode ter uma aparência como esta:

- *products/*
  - *widget/*
    - *specs/*
      - *specs.html*

Se quiser oferecer um link à página principal *widget*, a partir da página *specs*, utilize `../` para solicitar ao navegador que retorne ao diretório pai:

```
<a href="../widget.html">Widget Page</a>
```

Caso deseje retornar *dois* níveis de diretórios e oferecer um link para a página principal de produtos, a partir da página de especificações técnicas, formate o link da seguinte forma:

```
<a href="../../products.html">Product Page</a>
```

### Uso do link relativo à raiz

O processo de utilização de links relativos para movimentação dentro das pastas de um grande website pode ser traiçoeiro, ou ao menos intrincado. Outro tipo que pode ser usado nesses casos é o link *relativo à raiz*.

Aqui está como você utilizaria um link relativo à raiz na codificação de um vínculo partindo da página de especificações técnicas para a página principal de produtos mencionada no exemplo anterior:

```
<a href="/products/products.html">Product Page</a>
```

A barra significa o protocolo e o nome de domínio da URL, um tipo de abreviatura para links.

### Links para certos elementos dentro de uma página web

Você também pode oferecer links a certos elementos dentro de um documento HTML criando âncoras. Uma âncora pode ser criada atribuindo-se um atributo `id` a um elemento HTML:

```
<h2 id="hireme">Hire Me</h2>
```

Em seguida, ofereça um *link* para a âncora precedendo o nome `id` com um símbolo cerquilha (`#`):

```
<a href="#hireme">Hire Me</a>
```

Ao clicar sobre esse elemento, o navegador acessará a parte do documento que contém o nome `id` correspondente.



Caso um documento *não* seja mais extenso do que a janela de exibição do navegador, não haverá mudança que denote o acesso ao link ancorado.

Designers utilizam âncoras para criar uma tabela de conteúdo no topo da página, permitindo uma navegação rápida até outras partes do documento. Essa abordagem é particularmente útil em páginas web que contenham grande quantidade de conteúdo, como uma forma de ajudar seus usuários a evitar a paginação em excesso.

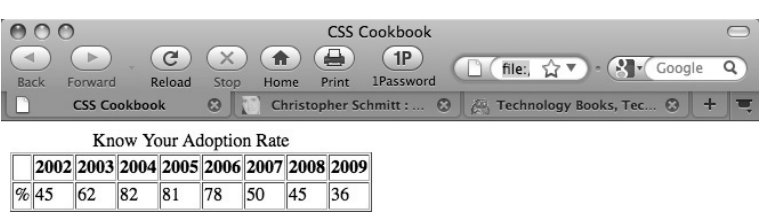
## Veja também

O capítulo 7 sobre links e navegação.

## 1.12 Codificação de tabelas

### Problema

Você deseja criar uma simples tabela em HTML, conforme mostra a figura 1.14.



The screenshot shows a web browser window with the title 'CSS Cookbook'. The address bar contains 'file:'. Below the browser window, a table is displayed with the caption 'Know Your Adoption Rate'. The table has two rows: a header row with years from 2002 to 2009, and a data row with percentages: 45, 62, 82, 81, 78, 50, 45, 36.

	2002	2003	2004	2005	2006	2007	2008	2009
%	45	62	82	81	78	50	45	36

Figura 1.14 – Renderização padrão de uma tabela HTML básica.

### Solução

Utilize elementos específicos relacionados à marcação de dados tabulares:

```
<table border="1" cellspacing="1" cellpadding="1">
  <caption>
    Know Your IE6 Adoption Rate
  </caption>
  <tr>
    <th>&nbsp;</th>
    <th>2002</th>
    <th>2003</th>
    <th>2004</th>
    <th>2005</th>
    <th>2006</th>
    <th>2007</th>
    <th>2008</th>
    <th>2009</th>
  </tr>
  <tr>
    <td>%</td>
    <td>45</td>
    <td>62</td>
```

```
<td>82</td>
<td>81</td>
<td>78</td>
<td>50</td>
<td>45</td>
<td>36</td>
</tr>
</table>
```

## Discussão

Primeiro, adicione uma tag `table` ao início e ao fim dos dados tabulares. Ela define a tabela como um todo.

O elemento opcional `caption` é utilizado para resumir os dados tabulares e aparece imediatamente depois do elemento `table` de abertura.

Em seguida, caso sua tabela contenha um cabeçalho, adicione a tag `thead` a uma ou mais linhas denotando o texto dessa seção. Utilize a tag `tbody` para envolver o corpo da tabela, distinguindo-o do cabeçalho.

Depois, adicione tags `tr` às linhas para marcar cada linha da tabela. Esse elemento engloba grupos de células individuais. Primeiro você define uma linha e depois adiciona as células internas.



Na codificação de uma tabela, não existem tags para colunas. É apenas pela criação de linhas sucessivas que elas surgem.

Após essa ação, utilize a tag `th` para cada célula que você deseja designar como título na tabela, as que incluem os anos e os percentuais na solução. Abrigue o conteúdo específico da célula dentro da tag. Por padrão, os navegadores exibem o texto das células de título em negrito.

Utilize a tag `td` para marcar células individuais em uma tabela. Assim como a tag `th`, a tag `td` envolve conteúdo específico das células.



Para um simples gerador de tabelas com base em HTML que contorne a criação manual de suas várias células, consulte <http://www.askthecssguy.com/kotatsu/index.html>.

## Veja também

O capítulo 9 sobre tabelas.

## 1.13 Criação de um vCard HTML (hCard)

### Problema

Você deseja incluir em uma página web suas informações para contato, de modo semelhante às que existem em um cartão de negócios, conforme mostra a figura 1.15.

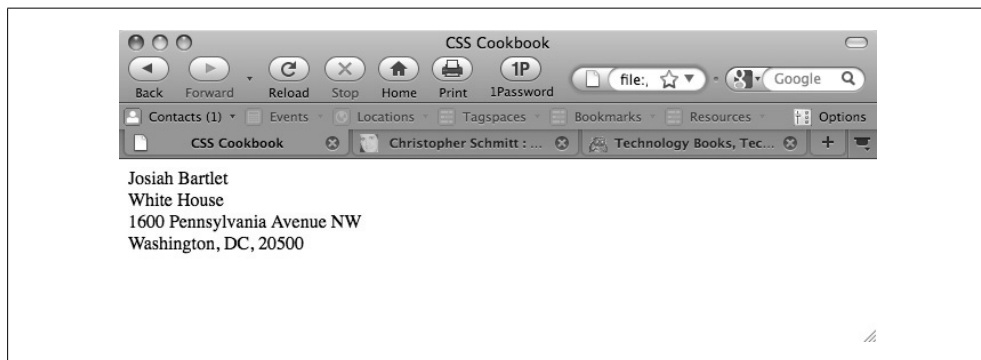


Figura 1.15 – Renderização padrão de um hCard.

### Solução

Utilize atributos `class` com atributos específicos listados na especificação do microformato hCard (<http://microformats.org/wiki/hcard>):

```
<div class="vcard">
  <span class="fn n">Josiah Bartlet</span>
  <div class="org">White House</div>
  <div class="adr">
    <div class="street-address">1600 Pennsylvania Avenue NW</div>
    <span class="locality">Washington</span>,
    <span class="region">DC</span>,
    <span class="postal-code">20500</span>
  </div>
</div>
```

### Discussão

O microformato hCard confere a você um modo de apresentar informações para contato, incluindo pessoas, organizações e locais, utilizando atributos de classe XHTML. É um dos muitos padrões detalhados no Microformats Project (consulte <http://microformats.org/>), cuja meta é fornecer um padrão para codificação de informações que sejam legíveis por máquinas, em páginas web utilizando HTML semântico. Assim como um padrão de design, um hCard padroniza a forma pela qual a informação é representada, permitindo que softwares de terceiros colem a informação e façam dela bom uso.

A fim de economizar tempo e evitar erros de digitação, utilize o hCard Creator (consulte <http://microformats.org/code/hcard/creator>) para gerar a sintaxe HTML.

### Extensão de hCards

O web service H2VX (<http://h2vx.com/vcf/>), que está disponível para uso no site e como um favelet, rastreia a marcação de uma página web, procurando por dados hCard a partir de um endereço web. Caso encontre um ou vários hCards, ele oferece a opção de baixar os dados na forma de um vCard.

O visitante, em seguida, pode importar o vCard para seu aplicativo de endereços preferidos, como o Outlook (Windows) ou o Address Book (Mac OS X).

O Operator (<https://addons.mozilla.org/en-US/firefox/addon/4106>) é um add-on para o Firefox que detecta texto microformatado em uma página, fornecendo várias opções de ação, dependendo do tipo de microformato utilizado.

Um plug-in semelhante está disponível para o Safari em <http://zappatic.net/safarimicroformats/>.

### Veja também

O validador de hCards, em <http://en.hcard.geekhood.net/>; e a receita 1.14 para utilização de HTML na marcação em código de um evento.

## 1.14 Marcação de um evento (hCalendar)

### Problema

Você deseja utilizar HTML para marcação de um evento.

### Solução

Utilize os atributos `class` e `title` com atributos específicos listados na especificação do microformato hCard (<http://microformats.org/wiki/hcalendar>):

```
<div class="vevent" id="hcalendar-The-CSS-Summit">
  <a class="url" href="http://csssummit.com/">
    <abbr class="dtstart" title="2009-07-18T09:00-04:0000">July 18, 2009 9</abbr>
    - <abbr class="dtend" title="2009-07-18T18:00-04:00">6pm</abbr>
    : <span class="summary">The CSS Summit</span>
    at <span class="location">Online Conference</span></a>
</div>
```

## Discussão

Com base no formato de arquivo iCalendar utilizado na troca de dados de eventos, o microformato hCard emprega HTML padronizado para codificar informações de horário e localização de eventos em um documento web.

Cada evento separado é designado com a classe `vevent`. Isso serve para especificar o conteúdo como uma entrada hCalendar.

O tempo de início do evento, `dtstart` e o `summary`, são *necessários* em todo evento hCalendar, enquanto as propriedades de término (end-time) `dtend` e `location` são opcionais.

Uma folha de referência para um hCalendar, disponível em <http://microformats.org/wiki/hcalendar-cheat>, oferece propriedades opcionais.

## Veja também

O hCalendar Creator (<http://microformats.org/code/hcalendar/creator>) e o Conference Schedule Creator (<http://dmitry.baranovskiy.com/work/csc/>) para criar facilmente seu próprio hCalendar; a receita 1.13 para inclusão de informações de contato em uma página web.

## 1.15 Validação do HTML

### Problema

Você deseja se certificar de que o HTML de sua página web está codificado corretamente.

### Solução

Utilize o validador do W3C (<http://validator.w3.org/>) para inserir a URL de um documento web e testar a validade de seu HTML, conforme mostra a figura 1.16.

De modo alternativo, você pode testar código enviando um arquivo CSS ou digitando as regras CSS.

### Discussão

O W3C emprega um robusto verificador de HTML em seu site. Contudo, a mensagem de resultado pode muitas vezes ser de difícil compreensão. Ao validar, certifique-se de habilitar a opção **More Options** (Mais opções) → **Verbose Output** (Resultado detalhado).

Essa opção de feedback fornece mais informações de fundo a respeito dos erros dentro de seu código, dando a você uma oportunidade maior de solucionar seus problemas.

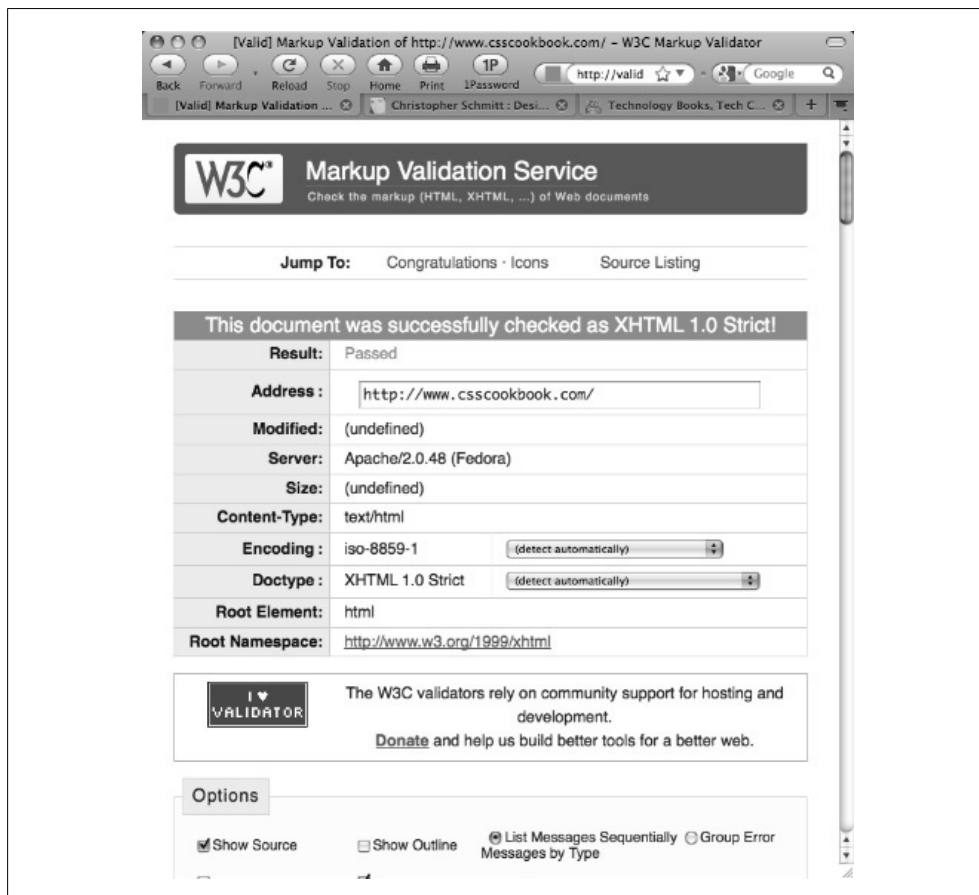


Figura 1.16 – Validação de uma página web.

## Criação de um bookmarklet validador de HTML

Leve qualquer página visitada na web diretamente ao validador de HTML do W3C utilizando um *bookmarklet*. Um bookmarklet é um pequeno segmento de JavaScript escondido na porção Address de um boCrie um novo bookmark, nomeie-o “Validador de HTML” e substitua o que quer que esteja no campo address pela seguinte linha:

```
javascript:void(document.location='http://validator.w3.org/check?
charset=%28detect+automatically%29&doctype=Inline&ss=1&group=0&
verbose=1&uri='+escape(document.location))
```

Ao visitar outro site, quando clicar sobre o bookmarklet você fará com que a página atualmente carregada no navegador seja submetida ao validador de CSS.

## Veja também

A receita 2.27 para validação de regras CSS.